EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

REVIEW Open Access

An empirical comparison of real-time dense stereo approaches for use in the automotive environment

Filip Mroz[1,2*] and Toby P Breckon[1*]

Abstract

In this work we evaluate the use of several real-time dense stereo algorithms as a passive 3D sensing technology for potential use as part of a driver assistance system or autonomous vehicle guidance. A key limitation in prior work in this area is that although significant comparative work has been done on dense stereo algorithms using *de facto* laboratory test sets only limited work has been done on evaluation in real world environments such as that found in potential automotive usage. This comparative study aims to provide an empirical comparison using automotive environment video imagery and compare this against dense stereo results drawn on standard test sequences in addition to considering the computational requirement against performance in real-time. We evaluate five chosen algorithms: Block Matching, Semi-Global Matching, No-Maximal Disparity, Cross-Based Local Approach, Adaptive Aggregation with Dynamic Programming. Our comparison shows a contrast between the results obtained on standard test sequences and those for automotive application imagery where a Semi-Global Matching approach gave the best empirical performance. From our study we can conclude that the noise present in automotive applications, can impact the quality of the depth information output from more complex algorithms (No-Maximal Disparity, Cross-Based Local Approach, Adaptive Aggregation with Dynamic Programming) resulting that in practice the disparity maps produced are comparable with those of simpler approaches such as Block Matching and Semi-Global Matching which empirically perform better in the automotive environment test sequences. This empirical result on automotive environment data contradicts the comparative result found on standard dense stereo test sequences using a statistical comparison methodology leading to interesting observations regarding current relative evaulation approaches.

1 Introduction

As automotive transport is one of the most important means of transportation in the modern world there is considerable interest in the advancement of both driver assistance systems, in order to improve both driving efficiency and safety, and the potential for autonomous driver-less vehicles. Dense stereo vision is a passive sensing technology offering a 3D view of the current environment making it a very attractive sensing component of such systems. In general such approaches consist of stereo matching between two on board camera sensors and the calculation of the scene depth map using triangulation based on the difference in relative position of objects/features in both camera images.

*Correspondence: fafafft@gmail.com; toby.breckon@cranfield.ac.uk
[1]School of Engineering, Cranfield University, Bedford, UK
[2]University of Wroclaw, Wroclaw, Poland

In general there are a range of proposed dense stereo methods [1-5] but the relative evaluation and comparative study of such techniques is limited [6-8].

Notable is the work of [6] which did not only provide a comparison of different algorithmic elements but also prepared a testing and ranking methodology where various algorithms can be easily and openly tested. In general this methodology highly stimulated the development of stereo vision algorithms but also focused attention on achieving good performance on a somewhat engineered image test set of static scenes. This test set is very different from the imagery occurring in the deployment of stereo vision systems in the automotive environment which we encounter in this study.

Later comparative work [7] has focused on real-time dense stereo algorithms. However this later study concentrated on a virtual automotive image sequence without the

noise and illumination problems clearly apparent in the real world automotive cases we consider here.

The problem with both of these studies [6,7] is that they use either artificially generated or artificially engineered non-automotive data for evaluation. The nature of this type of imagery can be significantly different from the real world imagery, such as that in automotive stereo deployment, which we consider here and contains various types of noise that can impact algorithmic results. In this study, it is thus necessary to use real world imagery in order to empirically evaluate the effect of such aspects on particular algorithms within a real world environment. It is also important to focus the evaluation around common dense stereo applications in real-world scenarios such as obstacle detection instead of the statistical pixel-wise compliance with test sets ground truth data. In this study it is driven by the expectation of the sensing capability of a driver assistance or autonomous vehicle subsystem.

By contrast the recent work of Klette et al. [8] relates its evaluation to the use of ground truth data but purports, as we do in this work *"the complexity of real-world data does not support the identification of general rankings of correspondence techniques on sets of basic sequences that show different situations."* Notably Klette et al. [8] concentrates on the wider aspects of stereo correspondence within the automotive environment and considers approaches achievable both in real-time and non real-time processing. The use of ground truth is limited and the ranking of algorithms that is performed is based on global statistical measures. Such measures are bias towards the correct disparity calculation of large textured scene regions (e.g. background) at the expense of the clarity of smaller, closer non-textured objects (e.g. other vehicles/pedestrians).

Here our study concentrates on comparing the performance of five chosen real-time algorithms [1-5] using data spanning from the Middlebury *de facto* test samples [6], synthetic data [7] and real world automotive stereo imagery. As has been previously reported [9] the performance of dense stereo algorithms highly depends on the imagery used and thus this comparison focuses on the factors that make these results differ in the translation from lab to real world scenarios. We cover a differing set of algorithms from [8], considering real-time requirements within the current state of the art, and base our evaluation, in this empirical study, on semantic scene interpretation in-place of global statistical analysis.

As we see in this study, the most significant and problematic feature of our real world automotive stereo data is into camera illumination variance which has to be limited for the dense stereo algorithms to work effectively. In this study we use a variation of Sobel operator [10] to remove this illumination problem but this can also affect overall algorithm performance due to noise amplification. This mutually supports the concurrent findings of [8] regarding future evaluation requirements.

From the five algorithms evaluated [1-5] we find that the best empirical performance was achieved by the Semi-Global Matching technique [2] which contradicts the results obtained over the Middlebury *de facto* test samples [6] and similarly differed from testing on the virtual automotive stereo imagery [7] where a Cross-Based Local Approach [4] empirically gave the most satisfactory results. This contradiction between real-world to basic test sample analysis further supports the findings of [8].

In this work we do not explicitly use ground truth comparison as in the comparative work of [8]. In general the use of ground truth data within real-time temporal (i.e. video) evaluation of stereo is limited to comparison against ground plain and simple background/foreground separation models [11,12]. Following that purported in [8] we look to the concept of semantic stability, as a conduit to the ready and reliable segmentation of foreground scene object (e.g. pedestrains/vehicles).

It is notable that with real world automotive stereo imagery the advantage of more complex and computationally expensive dense stereo techniques (requiring GPU computation for real-time performance [4,5]) is much less significant than in the case of the laboratory test set results [6]. We present a series of results comparing the performance of these algorithms on controlled environment stereo test sets [6], real world stereo imagery captured from an automotive stereo setup developed as part of this work and additionally from independent automotive stereo data [9].

2 Dense stereo vision

Stereo vision is a passive sensing technology for 3D measurement based on two perspective projections (i.e. camera images) of a given scene. This has been an active research area, with significant algorithm development over the past decades [6]. In dense stereo vision the distance (depth) to scene objects is calculated at each and every pixel location within the image. While this makes it computationally expensive, compared to its sparse stereo counterparts [13], recent increases in available computational power now make it possible within real-time bounds.

The general principle is to calculate the difference (here denoted as disparity) in the position of scene objects in both images (left and right) from which the scene distance (depth) is roughly inversely proportional to the disparity. Prior to this scene depth calculation the camera setup has to be calibrated to facilitate the *a priori* determination of various parameters characterising the stereo camera setup. In every 3D measurement cycle a pair of images (left/right) are first captured using synchronised cameras and then transformed to a standard geometry

(rectification) from which inter image pixel matching is performed and the resulting calculated disparity mapped to distance [14].

Calibration and rectification as well as disparity to distance mapping are well studied using established procedures [14-16] so here we assume that the input images are rectified (i.e. features aligned horizontally) and all that remains is disparity map (depth) calculation. Therefore our treatment of dense stereo is reduced to the problem of finding the dense correspondence between pixels in both left and right images.

Let D be the disparity map where $D(x, y) = d$ means that the pixel (x, y) in left image is matched with the pixel $(x - d, y)$ in the right one. Most methods require a restriction on this disparity range. In particular the maximum possible disparity which depends both on the cameras and the potential depth of the scene [1,2,4,5]. An alternative is given by the formulation of Unger et al. [3] where an iterative algorithm is presented that does not require such an assumption. We include examples of both such approaches in this evaluation.

The determination of the disparity map is formulated as a minimisation problem in which we look to minimise the overall matching costs between corresponding pixels. The best match D_{best} is defined [6] as the one minimizing this matching cost $E(D)$:

$$E(D) = E_{\text{data}}(D) + E_{\text{smooth}}(D)$$

where $E_{\text{data}}(D)$ is the sum of the matching costs between corresponding pixels and $E_{\text{smooth}}(D)$ is a penalising term for large disparity jumps within the disparity map, following the assumption that *"the physical world consists of piecewise-smooth surfaces"* [6].

The methods of finding an optimal approximation of D_{best} can be split into local and global methods. Global methods generally use one of the established optimization techniques (the best currently being graph-cuts [6]) to explicitly find D_{best}. However these methods are computational expensive and as such not currently an option for a real-time stereo requirement. By contrast local methods consider only a localised part of the image when approximating the disparity and minimise $E(D)$ only implicitly (by minimising $E(D)$ locally to obtain the best disparity D_{best}).

The excellent taxonomy-comparison-overview has been done by Scharstein and Szelinki [6] and similarly by Van der Mark and Gavrila [7] with a concentration on real-time methods. Both studies consistently identify the following breakdown of dense stereo matching approaches: preprocessing, pixel-based matching cost, cost aggregation, disparity search and post-processing. To this end we similarly follow this outline in our initial overview of dense stereo work in this area. Notably the preprocessing and post-processing are optional steps intended to improve the overall quality of the resulting disparity.

2.1 Preprocessing

The reality, that every image taken by real-world camera is noisy, poses a particular problem for stereo vision because of the assumption that objects in both the left and right images have the same visual appearance (i.e. color and brightness) characteristics. In reality this assumption is affected by noise, differences in camera characteristics and more significantly from problems arising from variations in illumination (and camera auto-gain response). This is a serious issue in case of real-time stereo imagery [17] as used in automotive applications.

The presence of simple Gaussian noise is commonly reduced using a median-filter [4] but to counter varying illumination additional filtering must be used. The common choice is a derivative operator such as Sobel, Laplacian of Gaussian (LoG) or residual image calculation [7,17]. All of these methods essentially calculate the derivative of the image signal (edge detection) and effectively eliminate illumination variance whose impact on the first and second derivative of the image is minimal. However these techniques tend to amplify image noise as a by-product of their use.

In the study here we make use of Sobel operator filtering in the horizontal (x-axis) orientation within the image, denoted as x-Sobel, to provide a horizontal first derivative gradient filter response as an input to all of the stereo algorithms considered. The raw Sobel operator output forms the input to the operation of pixel matching which we consider next.

2.2 Pixel-based matching cost

To perform matching we need to determine whether a given pixel (p_1) is similar to another (p_2) and define a quantifiable similarity measure. The constraint for such a measure, E, is that if p_1 corresponds (or is similar) to p_2 then $E(p_1, p_2) \approx 0$. A number of proposed solutions can be found in the literature [6,7]. Very common and simple approaches are the absolute difference (AD) $|p_1 - p_2|$ and squared difference (SD) $(p_1 - p_2)^2$ measures. An extension of these is a truncation of the result (e.g. $min(|p_1 - p_2|, T)$, with maximum threshold T) which limits the influence of one wrong pixel on the sum of the dissimilarities over the local area (section 2.3).

All the above measures are illumination and sampling dependent so they may not perform well when these differ between the left and right stereo images. In order to address this sampling problem Birchfield and Tomasi [18] presented a sampling invariant measure BT which is both computationally efficient and effective. The difference in lighting is a more difficult problem and if no preprocessing is done AD and SD fail severely. According to the recent work of Nalpantidis and Gasteratos [19], who propose a relatively fast lighting invariant method, as well as other sources [7], such measures are still beyond

real-time constraints. This supports the case where proper preprocessing of the incoming stereo image pairs is necessary prior to the application of one of the matching cost measures. These costs are then aggregated over the local neighbourhood.

2.3 Cost aggregation

If the matching cost over only a single pixel is used the resulting disparity will be heavily corrupted by noise. In order to improve robustness and also provide smoothness, local methods use supporting pixel areas (i.e. pixel neighbourhoods) to aggregate pixel-wise cost of a potential matching. The assumption is that all the pixels in the local area have the same disparity (i.e. are in the same distance) [6].

In the general case aggregated cost of the pixel (x, y) of having disparity d, denoted as $C_{xy}(d)$, is a sum of weighted costs over the support area (local neighbourhood):

$$C_{xy}(d) = \sum_{i,j \in S} E(L_{x+i,y+j}, R_{x+i-d,y+j}) W(i,j) \quad (1)$$

where L and R represents pixel positions in the left and right stereo images, respectively and W represents a weighting factor defined over the local pixel neighbourhood S indexed by neighbourhood iterators i, j.

The simplest case is when it is a square window with constant weight ($W(i,j) = 1$). The overall result depends on the size of the neighbourhood used and here we see a tradeoff between the prevalence of image noise and blur near object images within the resulting disparity.

A proposed solution is the use of adaptive shape areas chosen based on local colourise similarity to that of the anchor (i.e. centre) pixel (with assumption that the close surfaces of the same colour are at the same depth). A very good and fast example is proposed by Lu et al. [4] and Zhang et. al [20] where such an adaptive shape aggregation is done without increasing the complexity of the square window algorithm. An alternative method is based on adaptive weights corresponding to colour and distance of the supporting pixels to the anchor pixel. Good results, although non real-time, have been obtained using this method in [21]. A faster but still successful algorithm was presented in [5] where the support area was restricted to a vertical window.

2.4 Disparity search

Every possible disparity for a given pixel $\{C_{xy}(d)\}$ can be calculated the question remains how to choose the correct disparity, d, for a given pixel location xy from this set of disparities.

The simplest method is picking the disparity with the lowest associated matching cost, $C_{xy}(d)$. This approach is called Winner Takes All (WTA). The most significant disadvantage of WTA is that it does not cope well with untextured regions or the areas were repetitive features occur. In these cases the minimal matching value can be close to the value for some other disparities causing noise in the image to adversely affect the disparity search. In an automotive environment this is often the case because either the road, roadside features or vehicle surfaces are often featureless in general or repetitive in nature.

A more complex approach, yet still achievable within real-time is dynamic programming (DP). This technique optimises every scanline of an image separately, finding the set of disparities that minimise the overall scanline matching cost. It utilises a pixel ordering constraint which states that the ordering of pixels in both images is consistent.

The problem can be shown as finding a shortest path through the matching cost matrix (as illustrated in Figure 1). In Figure 1 we can see an example path where cell background colours (lower-bottom/right side) represent different pixel intensity values along the left and right scanlines respectfully. At every point internal to the matrix there are only three possible moves that correspond either to occlusion in the left or right image or a pixel match. As shown in Figure 1 from a given point (L_{occ} with red illustrative arrows) we can either move to the right (L_{occ}, indicating left pixel occlusion), upwards (R_{occ}, indicating right pixel occlusion) or diagonally (M_{RL}, indicating a match). A complete path thorough the matrix represents a match between the corresponding left and right scanlines in the images taking account of pixel matches and additionally pixels which are occluded in the left or right images respectfully.

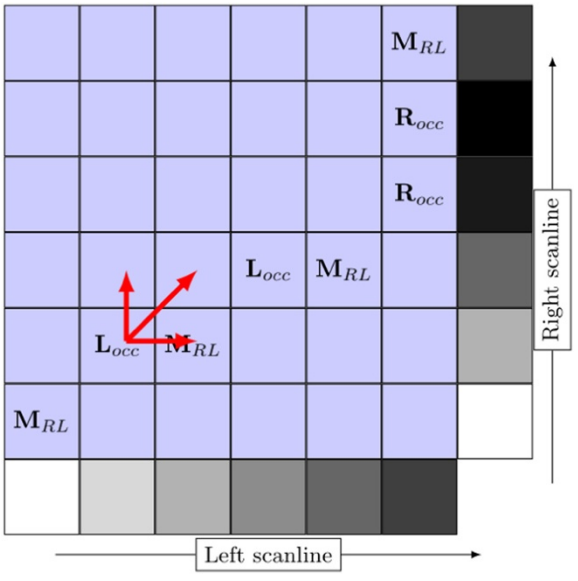


Figure 1 DP cost matrix example and an example path through it.

The advantage of DP is that it works well in untextured regions where there are no large depth discontinuities [7]. Within the DP formulation a left (L_{occ}), right (R_{occ}) pixel occlusion or a match (M_{RL}) has an associated cost. When a large disparity difference occurs in the image many pixels in the correct path will be occluded and the aggregated cost of such occlusions may outweigh the cost of an incorrect path without these occlusions. This results in wrong path being chosen though the DP cost matrix (Figure 1) and subsequently objects at given disparity being missed [7]. Another inherent problem is the lack of consistency enforcement between consecutive scanlines. However a number of solutions have been proposed to address this problem [5]. As identified in [7] these errors may interfere with subsequent steps of vehicle sensing (e.g the ground plane estimation) so a DP based disparity search is not ideal for this application.

Instead we concentrate on WTA approaches [1-4] and consider only one DP technique [5] in this comparative study.

2.5 Post-processing

Following pixel correspondence calculation some additional post-processing steps can be carried out to refine the resulting disparity image.

In the case of pixels occlusion in one of the image the calculation of the disparity is inherently incorrect and thus such areas have to be identified and filtered. Robust occlusion detection is provided by left-right consistency checking where the disparity is calculated twice (left to right and right to left) such that the resulting disparity map should thus only differ in occluded regions. Simpler and faster solutions look for a disparity gradient which indicates occlusions [3,4].

It is to be expected that some parts of disparity map, especially within the challenging automotive environment, will be incorrect. To identify these regions a disparity confidence measure is utilised. A sensible approach of [22] (see Equation 2) takes the two best matching costs (C_1,C_2), where $C_i = C_{xy}(d_i)$ for a given disparity d_i, and requires C_1 to be significantly lower than C_2 to ensure that the disparity decision has not been affected by noise (i.e. inter-confidence measure C_t is defined).

$$C_t = \frac{C_2 - C_1}{C_1} \tag{2}$$

Subsequently thresholding C_t to allow only high values results in a depth image of reliable pixel disparity. Due to sampling issues it may be the case that the correct disparity is actually between two consecutive pixels thus the matching cost for both pixels (C_1,C_2) may be similar causing in general C_t to be low. Such regions would be wrongly identified as unreliable disparity. The proposed solution of [7] is to maintain the best three matching cost values (C_i for $i = \{1, 2, 3\}$) and use the third one instead of the second (i.e. $C_2 = C_3$ in Equation 2). When the correct disparity is between two consecutive pixels the third matching cost value (C_3) must correspond to an incorrect disparity which if the pixel at (x, y) is reliable should yield relatively high matching cost resulting in high value of the confidence measure C_t.

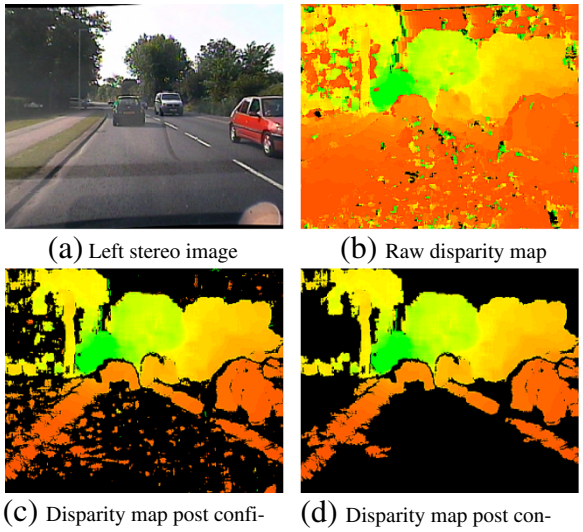
Another common post-processing approach is the use of the median filter that smoothes and removes irregularities [10]. The resulting disparity map often has significant speckle artefacts which can be subsequently removed using either segmentation or size based removal of disparity patches [2].

The affects of these two postprocessing on the disparity map are presented in Figure 2. Here we see that raw disparity map may have plenty of noisy (incorrectly computed) disparities especially in untextured portions of the image (Figure 2b). The proposed confidence check (Figure 2c, [22]) can remove most of these noisy disparities while the rest can be filtered by speckle removal (Figure 2d, [2]).

Finally, interpolation can be used to fill the holes caused by the previous refinements, providing dense or semi-dense disparity map [2].

3 Algorithm selection

The algorithm selection was intended to give coverage to most of the common-place stereo vision techniques thus the chosen five algorithms in this study represent a broad range of ideas within the field. Both major disparity selection methods (WTA and DP, see section 2.4) as well as different aggregation methods, such as fixed support area,



(a) Left stereo image (b) Raw disparity map (c) Disparity map post confidence check (d) Disparity map post confidence check and speckle removal

Figure 2 The effects of postprocessing of the raw disparity map.

adaptive shape support area and weighted fixed support area have representations in the final algorithm choice.

In this and subsequent sections the chosen algorithms are mentioned numerously. It is thus beneficial to use shorthand abbreviations and those in use are subsequently presented in Table 1.

In the following sections the chosen algorithms and their implementations are briefly presented.

3.1 Konolige block matching

Konolige [1] proposes a very simple and fast stereo vision algorithm. Its simplicity and regularity makes it possible to use efficient hardware optimisations to further enhance its speed.

The raw images are preprocessed with a LoG operator. The matching cost is then calculated as a sum of absolute differences (SAD) over a sliding square window (i.e. pixel neighbourhood) of fixed size. Finally the disparity is chosen using simple WTA approach.

In the efficient implementation used in this study the key difference is the usage of x-Sobel operator instead of LoG which calculates the derivative in the x-axis (horizontal). The run time does not depend on the SAD window size because the matching cost calculation utilises the fact that the neighboring windows overlap and thus it is implemented efficiently to add and subtract only the window differences.

3.2 Semi-global block matching

Semi-global block matching is a modified implementation of Hirschmuller's semi-global matching algorithm [2]. This algorithm is a recent attempt to achieve the quality of global method without violating real-time constraint. The idea is to approximate global cost function using a sum of 1D optimisations from all directions though the image.

For each of the eight directions (only five in our implementation) this 1D optimisation technique is used to calculate the matching cost. Each of these directions is solved using an approach similar to DP but in this case the ordering constraint (Section 2.4) is not enforced. The final matching cost is defined as a sum of these 1D matching costs and then the correct disparity is identified as the lowest matching cost (i.e. WTA).

One of the options presented in the original paper [2] uses mutual information (a sampling and illumination invariant measure) and a hierarchical setup but the available implementation utilises simpler BT measure (as discussed in Section 2.2). Strictly speaking this approach is not illumination invariant so x-Sobel preprocessing is added to preserve illumination invariance in the original algorithm.

Several disparity refinements are also discussed in the original article [2] but they are not present in the implementation utilised in this study. Instead confidence check and speckle removal are used as required as per Section 2.5.

Table 1 Algorithm abbreviation reference

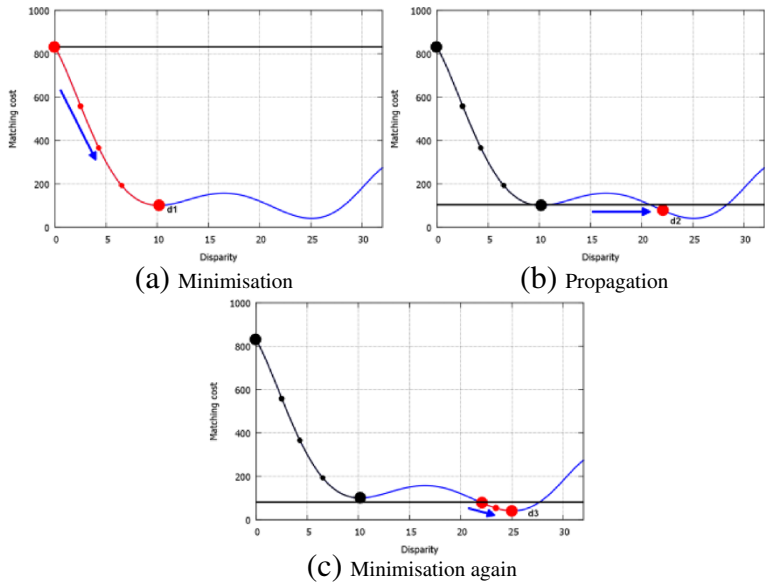
BLOCK	Block matching [1]
SEMI	Semi-global block matching [2]
NOMD	No-maximal disparity algorithm [3]
CROSS	Cross-based local algorithm [4]
ADAPT	Adaptive aggregation with dynamic programming [5]

3.3 No-maximal disparity approach

This method has been presented in [3] as a novel approach for stereo matching which does not require the explicit specification of maximal disparity (MD). In general, the choice of MD is important for the majority of stereo algorithms because if the chosen MD constraint is too small then we place an artificial depth limit upon the image meaning that objects with large disparity (i.e. close to viewer) cannot be found within the matching space. On the other hand, if the specified MD is too large the required computation increases, affecting the real-time performance, and the quality of the resulting disparity map can be effected [3]. The solution is to implicitly select MD for every pixel, by starting with a low value and increasing it independently for each pixel until the true disparity is reached. The work of [3] proves that this approach is successful on Middlebury data set [6]. In this study we verify if this result holds for real world automotive stereo imagery.

The proposed approach uses a hierarchical image pyramid which means that the disparity is first calculated for down-scaled images and then the resulting disparity is upscaled and the algorithm is reinitialised starting with this initial disparity approximation from the lower scale.

The matching cost used in this approach is a simple sum of the AD over a localised square neighbourhood. The iterative approach is proposed to terminate when the true disparity is reached and consists of two interlaced steps: minimisation and propagation (see Figure 3). Minimisation increases the current disparity until local minimum of matching cost function is found (Figure 3a, d_1). In the propagation step the disparity already calculated for the neighbourhood of the pixel is checked and if it yields the lower cost then the value is propagated to that pixel (Figure 3b, d_2). Subsequently the minimisation is performed again (Figure 3c, d_3). This simple idea prevents the disparity calculation from stopping in the first local minimum of the matching cost function. The remaining question arises of how many such iterations are required but it is stated in the original work [3] empirically only a few iterations tend to be required.



(a) Minimisation (b) Propagation (c) Minimisation again

Figure 3 The result of applying both steps on the matching cost function for a given pixel.

Simple pixel-wise refinement post-processing is additionally proposed to improve the disparity map after calculation. In general it is similar to propagation in the way that it propagates disparities through their immediate neighbours but this time it uses pixel-wise matching cost (AD or BT) and penalises disparity jumps thus smoothing the overall disparity image. In order to reduce the smoothing near edge features the penalty is reduced if the corresponding image intensity gradient is high.

In our implementation the AD measure is used for this post-processing step. The neighbourhood used in propagation step consists of left and right pixels but for the refinement it extends to the upper and lower neighbours of the pixel. We use the Sobel operator [10] to approximate the intensity gradient.

3.4 Cross-based local approach

The approach presented in [4] is a very good example of adaptive shape aggregation where shape depends on the colour similarity of the pixels. The key idea is to calculate, for every pixel, the size of a cross structure (horizontal and vertical lines) of similar pixels which is thus sufficient for further shape reconstruction and the matching cost aggregation.

In this algorithm the truncated AD is used as the similarity measure, $S()$. The first part is the calculation of the cross structure for every pixel in both images. Each of the four arms of the cross structure is defined as the longest sequence of pixels, starting from a specified anchor pixel, such that every pixel in the sequence is similar to the base pixel by measure S (see Equation 3).

$$S(p_1, p_2) = \max_{c \in \{R,G,B\}} |I_c(p_1) - I_c(p_2)| \tag{3}$$

Where $I_c(p_i)$ is intensity of channel c of the pixel p_i. The length of each arm $(h_p^-, h_p^+, v_p^-, v_p^+)$ of the cross is then truncated to fit into the interval of $[1, \mathrm{MAX_{ARM}}]$ where $\mathrm{MAX_{ARM}}$ is a parameter specifying the maximal arm length. These simple crosses fully define the support area for every pixel (see Figure 4). The area U_p (marked with red dashed line) consists of the horizontal arms h_q^* of the pixels lying on the vertical arms v_p^* of the anchor pixel.

During aggregation the step the support areas of both pixels are used symmetrically and the resulting aggregation region is defined as an intersection of both. Therefore region sizes can vary between different disparities so the average matching cost is used instead of a simple sum. The output disparity is the one with the lowest average matching cost with an additionally penalty in relation to the aggregation region being small in size.

In our implementation of this approach [4] the input images used in the calculation of the cross structures are always pre-filtered with a median operator and this is carried out prior to other specified preprocessing (i.e. x-Sobel).

3.5 Adaptive aggregation with DP

The approach of [5] uses adaptive weights in the aggregation step and a simple variant of DP for pixel matching. The colour similarity and the distance to the base pixel is used in the weight calculation. This weighting scheme, unlike the simple SAD, requires recalculation for every pixel which makes it computationally expensive [21]. Therefore the approach proposed in this paper

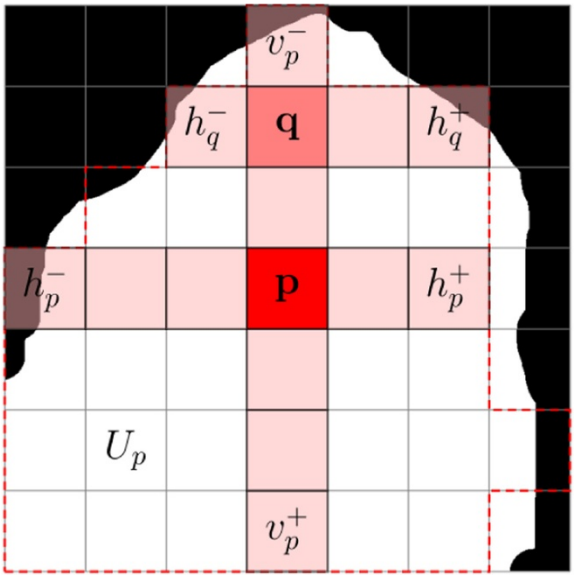


Figure 4 The support area is defined using the horizontal arms (h_q^*) of pixels belonging to the vertical arms (v_p^*) of the anchor pixel *p* [4].

limits the support area to a vertical window to achieve real-time capability. This restricted support area is still effective [5] against one of the main disadvantages of the DP—streaking effects. This is because the aggregation in use makes matching costs more consistent in the vertical direction.

The weight calculation is based on an exponential function. Let $w(p,l)$ denote the weight of l belonging to support area of pixel p. The $w(p,l)$ is then defined as follows:

$$w(p,l) = \exp\left(-\left(\frac{\Delta c_{pl}}{\gamma_c} + \frac{\Delta g_{pl}}{\gamma_g}\right)\right) \tag{4}$$

Δc_{pl} denotes intensity difference between pixels p and l where Δg_{pl} represents the Euclidean distance. These support pixel weights are subsequently used in the matching cost calculation which is expressed as a weighted sum $C(p_1,p_2)$ (Equation 5). S_i denotes the support area of pixel i for $i = \{p_1,p_2\}$. The weight is the multiplication of support pixel weights in both images.

$$C(p_1,p_2) = \frac{\sum_{l\in S_{p_1}, r\in S_{p_2}} w(p_1,l)w(p_2,r)\Delta c_{lr}}{\sum_{l\in S_{p_1}, r\in S_{p_2}} w(p_1,l)w(p_1,r)} \tag{5}$$

As with the previous approach (Section 3.4) input images to the weight calculation step are pre-filtered using a median filter.

4 Evaluation stereo data

In this study a range of stereo imagery is used for evaluation including that from the Middlebury Stereo Collection [6], virtual automotive stereo sequences [7] and also two sets of real-world stereo data originating both from this study and the prior independent study of [9].

4.1 Middlebury stereo test data

The Middlebury stereo data collection results from the prior stereo survey work of Scharstein and Szeliski [6]. It provides a wide range of stereo image samples with associated ground truth. In this study we select a subset of four basic examples for comparison. This reference stereo data set is established as a *de facto* standard test reference set for the evaluation of dense stereo algorithms. In this study it is used both to verify the implementations of our chosen algorithms and additionally provide a reference backdrop against our real-world study. Notably the Middlebury stereo data contains high contrast imagery with little to no illumination difference between the image pairs.

4.2 Virtual automotive stereo data

There are a number of artificially generated stereo imagery sets available including those applicable to automotive stereo vision. A representative one is that used for comparison study of Mark and Gavrila [7] which is a publicly available short video sequence. This data has additional distortions and noise added for enhanced realism but in general the overall signal to noise ratio is very low [7]. Nevertheless it is included in this study as a good example of representative automotive stereo data with the shapes and patterns commonly found in such an environment. The inclusion of this test data set in this study is to provide a halfway medium between that of the laboratory test sets (Section 4.1) and the real-world automotive stereo data (Section 4.3) to provide a greater depth to our evaluative results. Examples of this data set are shown in Figures 5 and 6.

4.3 Road environment stereo data

This stereo imagery set is definitely the most challenging with a number of potential issues that affect stereo correspondence: motion blur, camera synchronization, illumination variance and low image contrast. For the purpose of this comparison study two sources of real environment data have been utilised: independent stereo imagery from the Enpeda project [9] and stereo data captured in and around Cranfield, Bedfordshire, UK for the explicit purposes of this study.

4.3.1 Enpeda project imagery

The Environment Perception and Driver Assistance (Enpeda [9]) project provides a number of different stereo data sets.

Figure 5 Results for the city road view.

This stereo imagery contains more image noise and a significant increase in untextured scene regions that (compared to the stereo imagery of Sections 4.1 and 4.3) can be challenging to some stereo correspondence approaches. The illumination difference present in the imagery makes straightforward pixel matching challenging. However, in contrast to the stereo data from our capture (Cranfield University stereo imagery) there are no reflections visible as the stereo rig was mounted outside of the vehicle. In general the overall level of image noise is lesser than in the case of the Cranfield University stereo imagery.

4.3.2 Cranfield stereo imagery

In addition to the publicly available data sets for automotive stereo vision [7,9] local capture was also performed using a stereo rig explicitly constructed for this study. The stereo rig carrier vehicle was a Ford C-MAX saloon car (2010) (Figure 7a). The internal view of Figure 7b shows the internal mounting of two analogue output CCD type cameras (camera: Sony 1/3″ HQ1 CCD / X-Vision IXC1HQDNE DSP) on the windshield using a dual camera suction mount. Despite shielding some mild windscreen reflection and illumination variance between the images was suffered. This was compensated for by the x-Sobel preprocessing utilised in all of the stereo approaches evaluated.

The range of stereo data captured includes environments such as the rural university campus, village and town environments as well as open road and dual carriageway (multi-lane highway).

5 Evaluation and discussion

Our evaluation, in contrast to the earlier work of [8], considers the semantic interpretation of the resulting disparity map in terms of object cohesivity, connectedness and separation within the environment.

As earlier works have shown the alternative of ground truth evaluation is limited to global statistics [8] over sparsely labelled ground truth (e.g. [11,12]). This is inherently bias to the corrected of the (often highly textured) scene background. Here we attempt to qualitatively evaluate the ease of semantic disparity image interpretation with a bias towards the clarity and temporal stability of scene foreground objects (e.g. pedestrians/vehicles/street furniture). It is after all these which present key challenges for future driver assistance systems using stereo sensing.

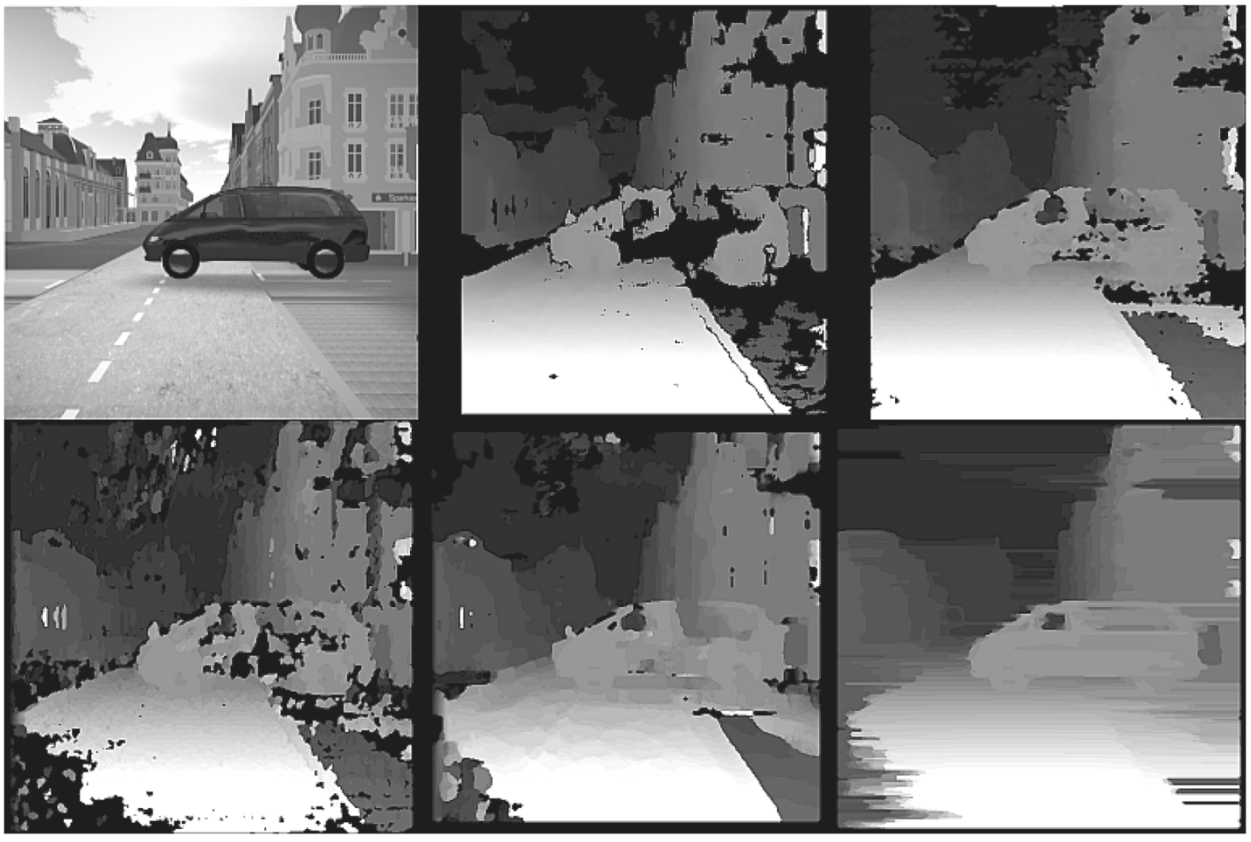


Figure 6 Results for the crossing vehicle.

Qualitatively evaluating such foreground objects within such sequences is an area for future work.

As many visual comparisons of our five chosen algorithms are presented throughout this chapter as figures, we present a permanent layout key of the resulting depth maps in Table 2. The names of the algorithms relate back to the key presented earlier in Table 1 and are numerated from one to six as per the bracketed numbers.

We present a range of Figures (Figures 8, 9, 5, 6, 10, 11, 12 and 13) as appropriate stills from the sequences to illustrate our evaluation criteria and in addition a corresponding set of videos available at "http://www.cranfield.ac.uk/∼toby.breckons/demos/autostereo/" to further illustrate any temporal aspects.

(a) The carrier vehicle (b) Internal view

Figure 7 Capture equipment.

5.1 Common experimental method

Furthermore a few initial remarks about the choice of input parameters for the algorithms have to be made for clarification at this stage. All of the five algorithms except for CROSS use grayscale images as the input to disparity calculation. Whereas the other algorithms could have been adapted to use a colour input only CROSS used an RGB colour input in the original work [4]. BLOCK and NOMD both use the same SAD window size in order to prevent this difference from influencing the comparison.

The important part of any stereo algorithm is the preprocessing stages especially when the illumination invariant filters are utilised (x-Sobel). The use of such a filter greatly affects the resulting disparity map and such effects

Table 2 Layout of the depth maps presented in Figures 8, 9, 5, 6, 10, 11, 12 and 13

Input left	BLOCK [1]	SEMI [2]
NOMD [3]	CROSS [4]	ADAPT [5]

Figure 8 Tsukuba image pair.

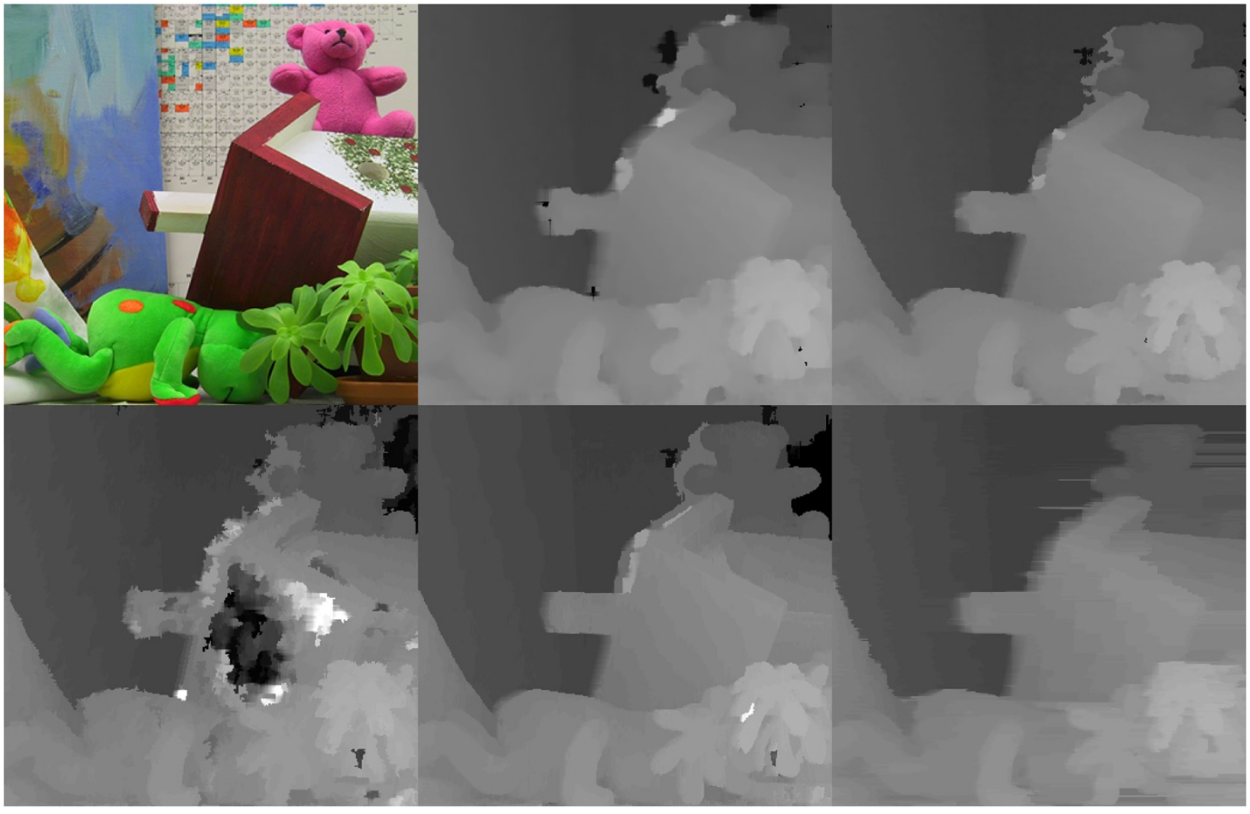


Figure 9 Teddy image pair.

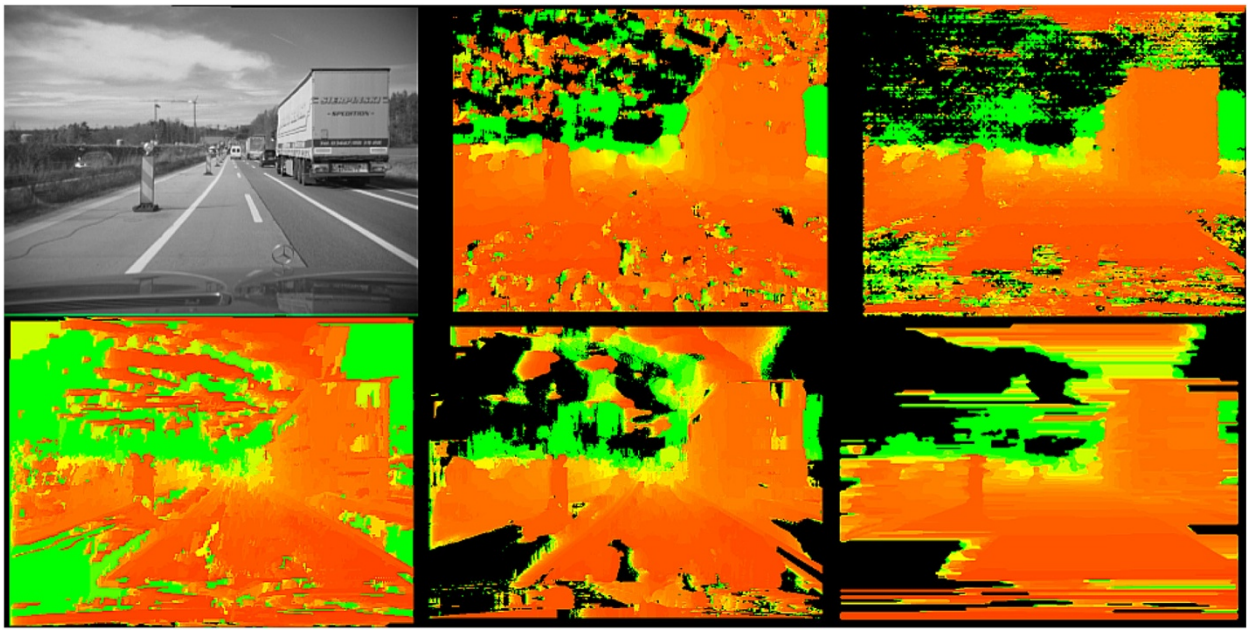


Figure 10 Results for the preprocessed input (BLOCK and SEMI) and *x*-Sobel raw input (NOMD,CROSS and ADAPT).

are to be detailed in this study. In the case of BLOCK and SEMI the implementation used inherently uses *x*-Sobel filtering so unlike the other approaches (NOMD, CROSS and ADAPT) we are unable to investigate how this filtering affects the resulting disparity map (compared to using raw input).

The post-processing step utilised in this evaluation includes confidence check and speckle removal for all the approaches except for ADAPT in which case no post-processing is performed. This is because it is not possible to adapt the confidence check to consider whole scanlines rather than separate pixels.

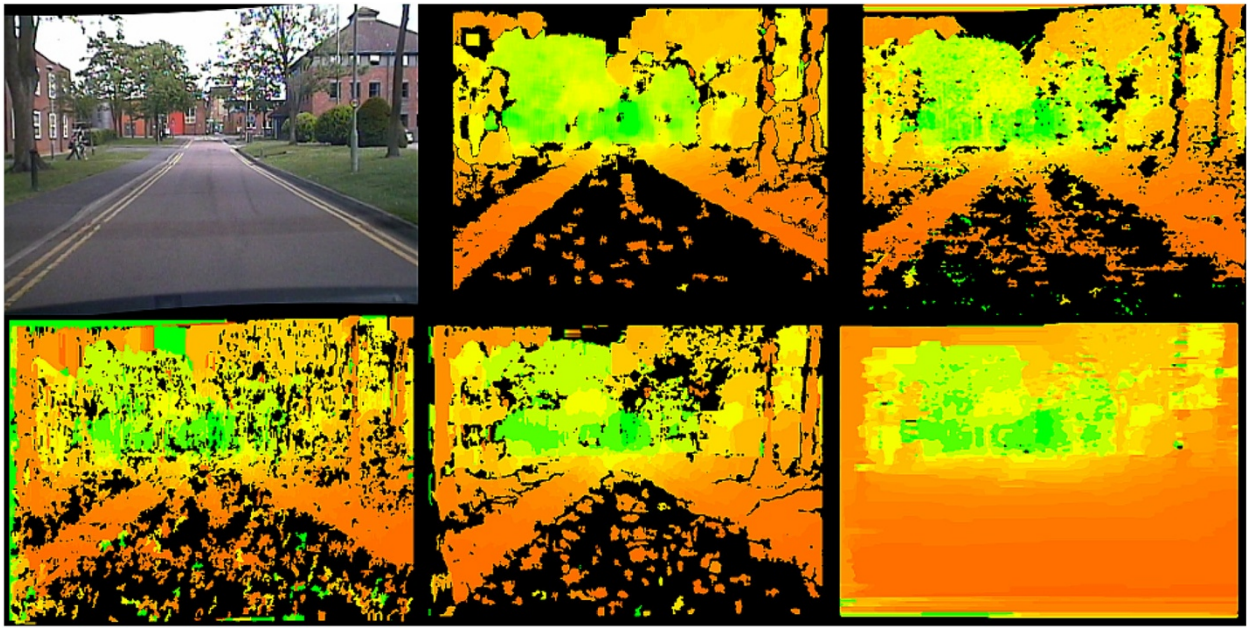


Figure 11 University campus road.

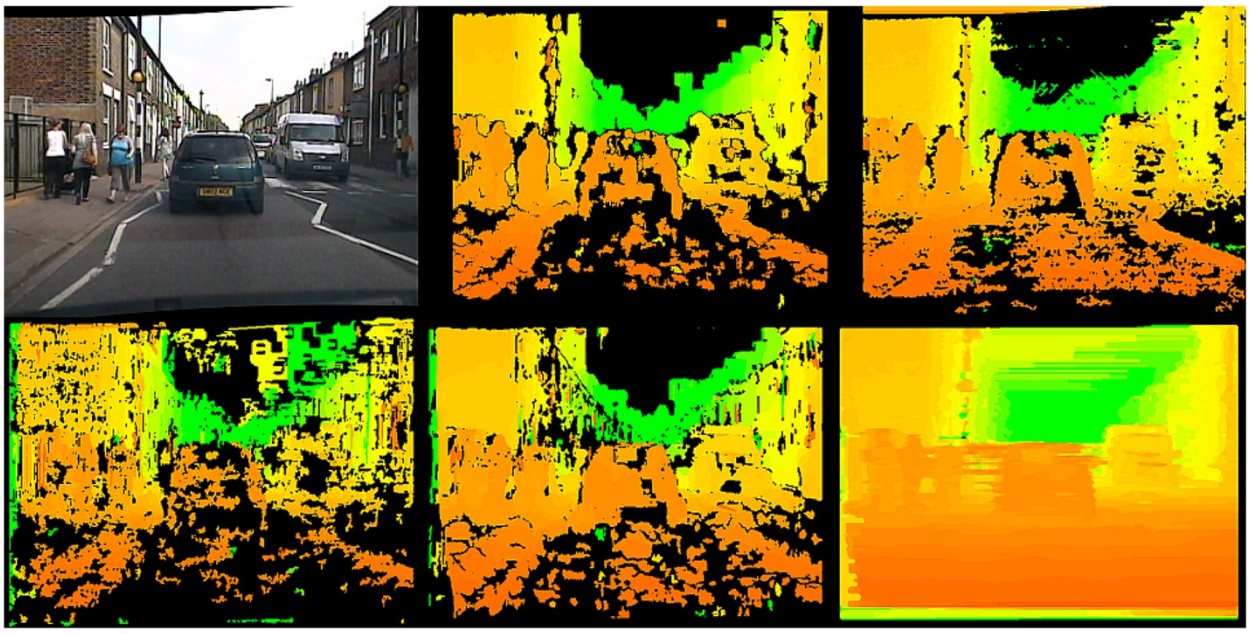


Figure 12 Town view.

All five algorithms are evaluated on the same test imagery sequences.

5.2 Middlebury stereo test data

These *de facto* stereo test data sets are used to verify algorithm performance against ground truth in the presence of low image noise and in many ways represent an experimental control for our work.

The algorithms were tested with one fixed set of parameters for the four standard stereo pairs with a numerical results detailed in Table 3. The numerical value in the table represents the percentage of pixels that have been

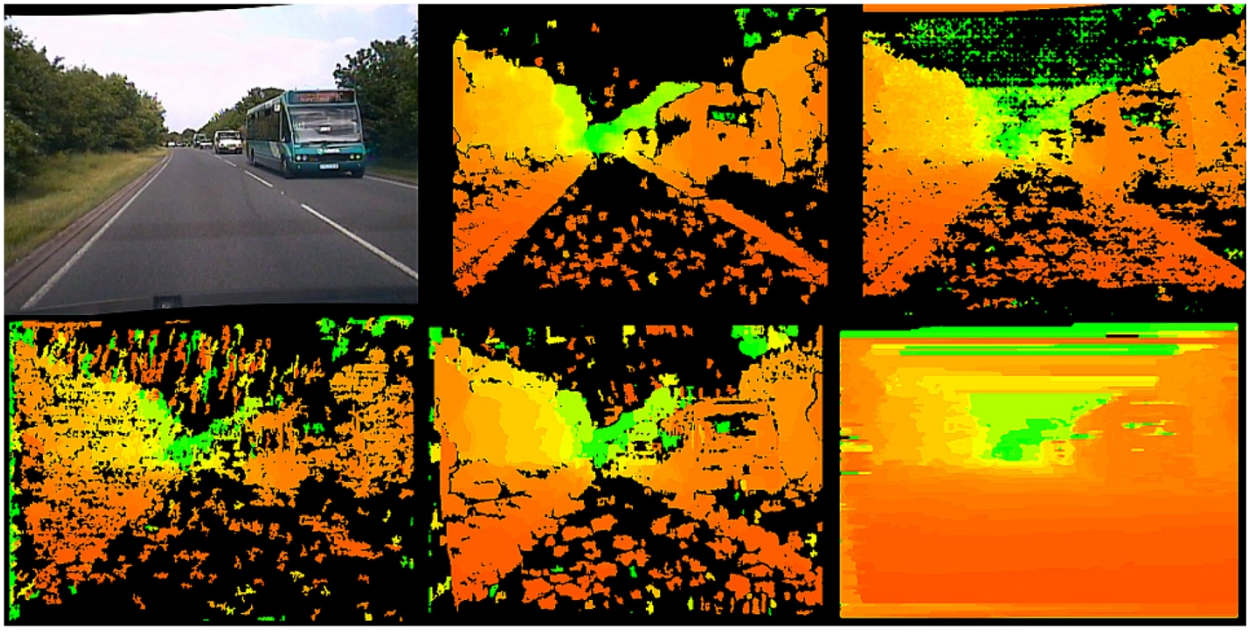


Figure 13 Open road view.

Table 3 Middlebury stereo data sets results

	Non-occluded bad pixels [%]				
	Tsukuba	**Venus**	**Teddy**	**Cones**	**Average**
BLOCK	6.47	2.96	8.85	6.04	6.08
SEMI	3.92	0.85	4.08	2.97	2.96
NOMD	6.93	6.68	17.92	13.62	11.30
CROSS	3.42	2.09	5.49	5.08	4.02
ADAPT	4.32	2.92	7.09	8.42	5.69

incorrectly calculated by a specified algorithm in comparison to the *a priori* ground truth [6] not including any occluded regions.

Table 3 shows that SEMI is by far the best performing but CROSS results are also good. The surprising fact is that NOMD results are very poor. The visual results for Tsukuba and Teddy data sets are presented (see Figures 8 and 9) for further investigation.

CROSS [4] is arguable the best in terms of computing the disparity near the edges of the objects (see the lamp in Figure 8) which is because of the adaptive shape of the aggregation region. SEMI [2] is also able to perform well but due to the x-Sobel preprocessing (Section 2.1) objects edges are not as precise. The drawback of CROSS is that it completely fails in regions with no texture (e.g. the area on the right of the pink teddy bear—see Figure 9).

The NOMD [3] algorithm seems to have an issue with the refinement step because although a number of variants have been tested it was not possible to precisely replicate the results presented in the original paper [3]. For instance the wooden roof in Teddy is computed incorrectly because of repetitive texture which is an issue for every WTA algorithm. If the parameters of the refinement are changed so that the problem is fixed then the disparity from the table in Figure 8 is erroneously propagated into the shadow beneath it. Although BLOCK [1] also uses WTA to choose disparity it succeeds in these parts of the Teddy image due to x-Sobel preprocesing enhancement of the texture information. The usage of fixed support area size in BLOCK results in the significant blur near the edges and the disappearance of some of the small features (e.g. thin parts of the lamp in Figure 8).

ADAPT [5] as a representative DP approach does not have a problem with the repetitive texture in Figure 9 but it fails in the disparity jumps next to the teddy bear. The vertical aggregation is limiting the horizontal streaking effects.

Overall we see foreground object cohesivity, connectedness and separation across all of the five algorithms that lends itself well to the manual (or automatic) semantic intrepretation of the disparity image in isolation. Based on these results we now contrast these finding to the use of these same five real-time dense stereo algorithms on firstly virtual automotive test data and secondly on real automotive environment stereo imagery.

5.3 Virtual automotive stereo data

This virtual automotive data set [7] has an important set of application specific features and objects such as vehicles, buildings and the road surface with associated texture. This significantly differentiates them from the previous data set whilst in the same time allowing evaluation of the algorithms performance in a virtual environment without noise related issues.

The visual results in Figures 5 and 6 present two different points within the virtual test sequence. The first one illustrates a clear road between the buildings whilst the other representing a vehicle approaching a junction. The resulting disparity maps have been post-processed using confidence check and speckle removal in order to filter out noise for all five algorithms under consideration.

As can be seen in Figures 5 and 6 all of the approaches perform reasonable well upon this data set although CROSS [4] is again seen as the best performing algorithm in terms of fine depth detail, foreground object connectiveness and cohesitivity and limited disparity noise. The problem of this approach is that its assumption that close objects of similar colour are at the same distance is violated by the automotive environment features present in this stereo image pair (e.g. at some distance lanes merge into one white line). The repetitive pattern on the sidewalk of the example imagery (Figure 5) is clearly something that WTA based approaches (even SEMI [2]) cannot readily cope with and this results in erroneous disparity calculation in this area.

Figure 6 highlights the fact that vehicles within such imagery usually have no strong texture on their surface and thus as the distance to the camera decreases there is a probability that the resulting disparity image splits the vehicle into a number of smaller depth segments. This would make the potential use of vehicle identification from stereo in driver assistance systems challenging. From the results of Figure 6 we can see that ADAPT [5] is inherently resilient to this situation but out of all of the WTA approaches CROSS [4] seems to outperform the others from a subjective view point. This can be largely attributed to the fact that in such a case the aggregation region is large so thus it is more likely to match to the correct region in the corresponding image.

Overall from initial testing on the virtual automotive data set of the two isolated scenarios we can see the varying performance of the algorithms in the virtual test sequence. The performance of these algorithms is somewhat similar to that encountered within the area of Middlebury test sets but we identify significant potential issues with use in a typical varying texture automotive environment. Notably, we see differing performance

between the general scene background and forground object cohesivity in terms for suitability for effective foreground object segmentation or recogniton. This analysis differs from the global statistical evaulations of prior work [6,8] where the relative performance of the algorithms on the scene background would dominate. Again videos of this work are available from "http://www.cranfield.ac.uk/~toby.breckons/demos/autostereo/".

We move forward from these results to consider real automotive environment stereo imagery.

5.4 Road environment stereo data

In this section we present the evaluation of the chosen stereo algorithms on real automotive environment stereo imagery based on two data sets; (1) Enpeda project stereo imagery [9], (2) Cranfield University stereo imagery.

5.4.1 Enpeda project imagery

In this data there is no windscreen reflection present due to the camera rig being mounted on the exterior of the vehicle. In Figure 10 we show the results for an illustrative image pair for further investigation.

In this set of calculated disparity maps (Figure 10) the best performing approach is clearly SEMI [2] because both the close and distant objects within the scene are very differentiable. In Figure 10 only the upper right (BLOCK, SEMI) algorithms have been preprocessed with x-Sobel filtering. However, Figure 14 shows that the x-Sobel filter preprocessing considerably improves the performance in the case of the lower three algorithms illustrated in this figure (NOMD, CROSS, ADAPT).

The main improvement (with x-Sobel, Figure 14) occurs in the areas of low texture such as the back of the truck and the road surface. Both ADAPT (Figure 14, right) and CROSS (Figure 14, middle) are now able to correctly estimate the near road surface disparity. However, both with and without x-Sobel operator, in general CROSS performs worse than simple BLOCK (see Figures 10 and Figure 14). This can be illustrated in the fact it fails to determine the second roadside sign visible in the right middle of the scene. Additionally we can note the amplified noise in the x-Sobel input images also affecting ADAPT approach—as we can see the top of the first roadside sign is missing (Figures 10 and Figure 14).

The speckle effect visible in the case of NOMD (Figure 14 left) can come either from the final refinement stage which uses pixel-wise costs or the usage of the iterative approach. In order to ascertain the impact of the refinement stage we compare the resulting disparity maps with and without this stage in computation of NOMD approach (Figure 15).

As we can see in Figure 15 the horizontal speckles are clearly visible before applying the refinement stage thus it is the iterative setup of this approach that originally introduces them. The shape of these speckles is ultimately caused by the suggested NOMD methodology that optimises each row separately (Section 3.3). As we can see with refinement deactivated the iterative approach is the only feature that separates NOMD from BLOCK and we can thus see it is clearly failing when x-Sobel preprocessing is used as a illumination invariant pre-filter.

Finally Figure 16 presents the disparity maps of all five approaches when the input imagery is preprocessed with x-Sobel filtering and the resulting disparity maps are post-processed with confidence check and speckle removal (see Section 2.5). In a similar vain to the earlier results we can clearly see that in terms of depth consistency and object separation again the SEMI outperforms the others. Additionally we can see that the post-processing step removed most of the noisy disparities including the speckles in the disparity map produced by NOMD. Notably Figures 10, 14, 15 and 16 additionally illustrate the varying performance on forground object detail as previously encountered.

5.4.2 Cranfield university automotive stereo imagery

We now move on to consider stereo imagery that contains both mild camera to camera illumination variance and additionally mild windshield reflections from the internally mounted cameras.

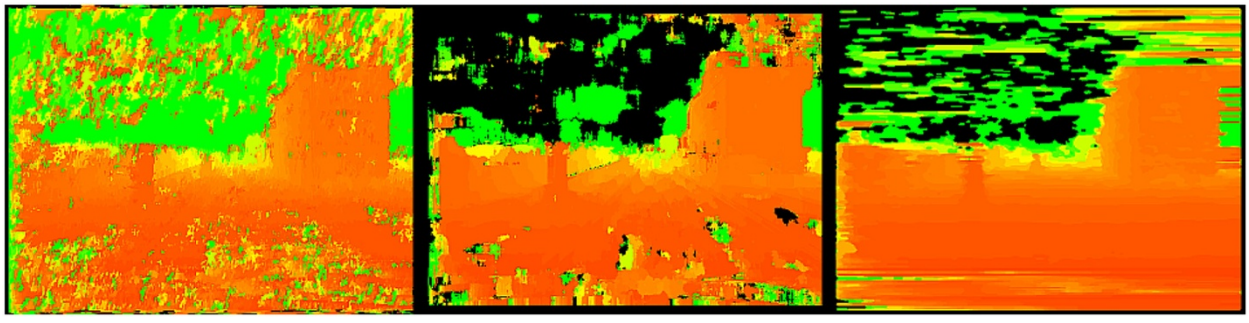


Figure 14 Results of NOMD (left), CROSS (middle) and ADAPT (right) for the x-Sobel preprocessed input.

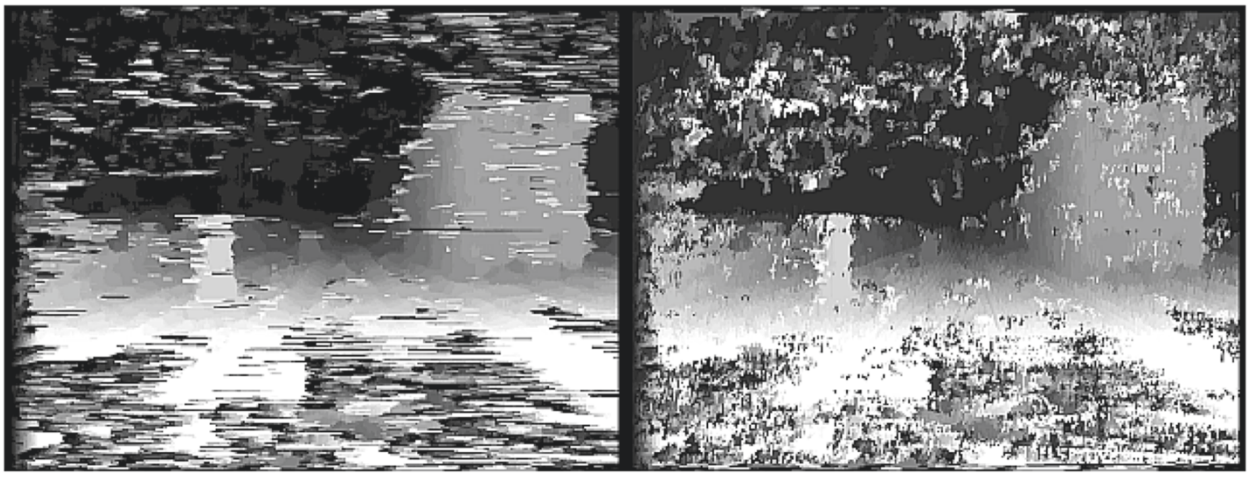


Figure 15 The raw NOMD disparity maps without refinement (left) and with refinement (right).

We investigate the results of the chosen techniques over various automotive scenes. All of the presented disparity maps (in Figures 11, 12 and 13) have been post-processed with confidence checking and speckle removal (Section 2.5) to eliminate disparity noise.

The first image test set (Figure 11) is a university campus road scene with a number of trees and poles as well as a group of pedestrians on the left hand side. All the approaches have correctly identified the pedestrians and thin vertical obstacles (i.e. trees and poles). The only exceptions is ADAPT [5] which due to its DP origin partially missed close objects (i.e. pole and tree on right). BLOCK [1], SEMI [2] and CROSS [4] all performed similarly with SEMI showing marginally superior performance on the distant tree disparity calculations. However this difference is not significant which in itself is a notable result. The pedestrian and vehicle forground objects are notable more readily separable in the BLOCK and SEMI approaches.

The similarity between the BLOCK [1] and CROSS [4] can be explained by the fact that if similarity threshold (defined in CROSS, Section 3.4) is large then the adaptive aggregation neighbourhood is larger and, in some regions of the image, not limited by the pixel similarity

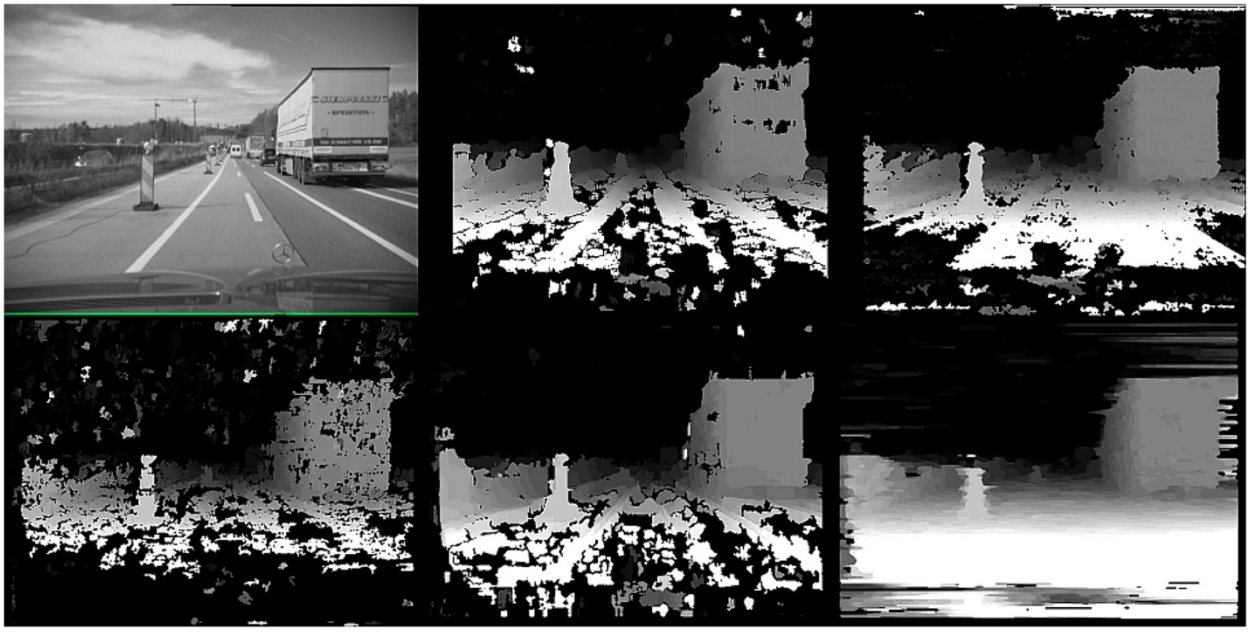


Figure 16 Post-processed results for the *x*-Sobel preprocessed input images.

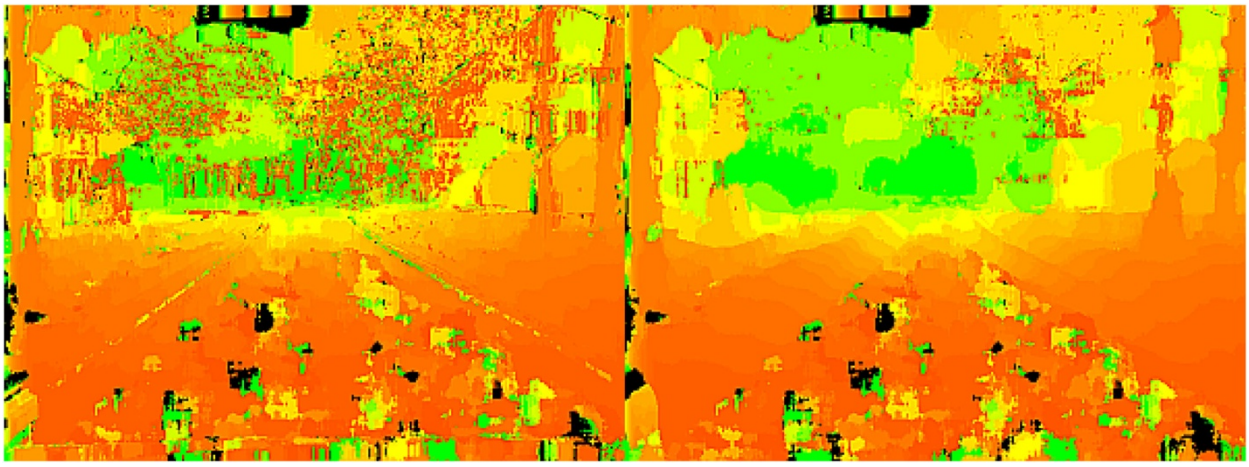


Figure 17 Disparity maps calculated by CROSS with lower (left) and higher (right) similarity threshold.

but by MAX_{ARM} value (Section 3.4) which results in the square shape aggregation neighbourhoods (the same as in BLOCK). As a result different values of this similarity threshold (Section 3.4) are tested for comparison with the results shown in Figure 17.

From Figure 17 we can observe that the amount of noise in the left disparity image (lower similarity threshold) is significantly greater than the right image (higher similarity threshold) which can be interpreted as meaning that the smaller aggregation regions cannot readily cope with the noise present in the image (which is itself amplified by x-Sobel pre-filtering). In this case we are thus forced to use a higher similarity threshold value causing BLOCK and CROSS to appear comparable in terms of disparity map results.

This result of the comparable disparity maps for these three algorithms hold for the other stereo samples from this set. Figures 12 and 13 present two substantially different scenes (town view and open road) but the observations are similar for both of them with the disparity map clarity and consistency of SEMI subjectively outperforming that of the other algorithms (notably in the relative stability and separability of foreground objects).

Notably ADAPT [5] has again problems with disparity jumps which is exposed in Figure 12 where the group of people on the left merge with the car in the foreground. In Figure 13 the disparity from the distant trees propagates into the untextured sky. In terms of CROSS [4] versus BLOCK [1] comparison the former works better in some situations (e.g. leading vehicle in the town scene, Figure 12) whilst in others (probably when the aggregation region is small) the noise causes erroneous disparity calculations (e.g. left road side post, Figure 12). In spite of the speckled appearance SEMI [2] is capable of reliably recognising all the thin vertical features of the scene (Figures 12 and 13). Nevertheless it is interesting that there is no substantial difference between the SEMI [2] and BLOCK [1] which is significantly different from the observation of Section 4.1 where the algorithms were tested against the *de facto* Middlebury stereo imagery.

From the analysis of this section we can conclude that when the noise level increases withing the imagery (i.e. Cranfield stereo imagery) the difference in performance between the most and least complex algorithms decreases in general. For use within the real world automotive environment usage of more sophisticated, computationally intensive algorithms may be largely unjustified because the increase in resulting disparity map quality is limited if it improves at all. Furthermore this is supported from the testing carried out on the independent stereo data set from an externally mounted camera rig of the Enpeda project [9]. It should be noted that these results contradict the relative performance shown against the Middlebury *de facto* stereo imagery test set (Section 4.1). The examples presented in this section, in comparision to the earlier virtual automotive stereo data, are made available as videos at "http://www.cranfield.ac.uk/∼toby.breckons/demos/autostereo/" for further consideration of temporal consistency and conhesitivity.

Table 4 Summary performance of the evaluated real-time dense stereo approaches

Approach	MDE/s	320 × 240 [fps]	Hardware
BLOCK [1]	351	190.4	CPU: 1.5 GHz
SEMI [2]	47	25.5	CPU: 1.5 GHz
CROSS [4]	30	16.3	GPU: GF7900GTX
NOMD [3]	–	25.8	CPU: 2.67 GHz
ADAPT [5]	50	27.1	GPU: XL1800

5.5 Real-time processing capability

This work does not directly focus on the processing speed of the evaluated approaches but instead on quality of the 3D disparity map produced. We use the information of the original works [1-5] to sufficiently conclude that these algorithms can perform in real time (notably with some requiring generalised GPU support [4,5]).

In Table 4 we present unified run time information from the original works [3-5] and for BLOCK/SEMI the results were measured from those of this study. MDE/s stands for million disparity estimations per second and is used as a single parameter characterising the overall performance of a given stereo algorithm. For BLOCK, SEMI, CROSS this was calculated from the run time on a given image and disparity range. The number of processed frames per second was then calculated assuming disparity range of 24. The NOMD presented a problem because the method does not inherently depend on a specified disparity range which is required for MDE/s calculation. Therefore the predicted run time is computed as the average time spent to processed one image pixel (denoted as $k = \frac{\text{runtime}}{\#\text{pixels}}$) based on the information of [3] then scaled by the number of pixels in the image and inverted to retrieve the number of processed frames per second as presented in Table 4.

6 Conclusions

A number of important conclusions emerge from the results presented in this work. Firstly Semi Global Matching (SEMI) [2] performs the best in almost every aspect of the disparity calculation however the difference between it and much simpler and faster Block Matching (BLOCK) [1] is not substantial. Furthermore the disparity maps of all the WTA approaches [1-4] are roughly similar while in the case of the simple stereo data of Middlebury the difference was very significant. The difference that we see in all of these approaches is that the noise in automotive data makes the performance very different from the statistical comparison of the *de facto* Middlebury test set [6]. Furthermore, from our analysis of relative real-time performance, it is questionable as to whether the quality gain in the automotive environment from BLOCK to SEMI is worth the additional computational effort. Prior work in automotive stereo has similarly considered such algorithm comparison in statistical terms [8] alone whereas here we present an empirical study based primarily on the requirements of foreground object separation/detection requirements for use in driver assistance systems against the backdrop of required computational effort.

Overall we have clearly identified a significant difference between the laboratory condition stereo imagery results obtained on the statistical evaluation of the *de facto* Middlebury stereo test imagery [6], the virtual world automotive stereo imagery [7] and that achieved under real world automotive stereo conditions. This difference between laboratory test conditions and the deployment of the stereo algorithms in the real world automotive environment should be considered for stereo use in applications such as obstacle detection, vehicle guidance and driver assistance systems. Prior work on statistical evaluation methodologies [6,8] is biased towards good performance on large scene areas (i.e. background) at the expense of important foreground objects.

Future work will investigate the inclusion of further algorithms and also consider common variations in weather conditions typical of the automotive environment in addition to quantitative methodologies for stereo algorithm evaluation that overcome the background bias identified in [6,8].

Competing interests

The authors declare that they have no competing interests.

Received: 16 March 2012 Accepted: 3 July 2012
Published: 16 August 2012

References

1. K Konolige, in *International Symposium on Robotics Research*. Small vision system, hardware and implementation, (1997), pp. 111–116
2. H Hirschmullerm, Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 328–341 (2008)
3. C Unger, S Benhimane, E Wahl, N Navab, in *British Machine Vision Conference*. Efficient disparity computation without maximum disparity for real-time stereo vision, London, 2009), pp. 42.1–42.12
4. J Lu, K Zhang, G Lafruit, F Catthoor, in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*. Real-time stereo matching: a cross-based local approach, (2009), pp. 733–736
5. L Wang, M Liao, M Gong, R Yang, D Nister, in *Third International Symposium on 3D Data Processing, Visualization and Transmission*. High-quality real-time stereo using adaptive cost aggregation and dynamic programming, (2007), pp. 798–805
6. D Scharstein, R Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vis. **47**(1-3), 7–42 (2002)
7. WVD Mark, DM Gavrila, Real-time dense stereo for intelligent vehicles. IEEE Trans. Intell. Transport. Syst. **7**, 38–50 (2006)
8. R Klette, N Kruger, T Vaudrey, K Pauwels, M van Hulle, S Morales, F Kandil, R Haeusler, N Pugeault, C Rabe, M Lappe, Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database. IEEE Trans Veh. Technol. **60**(5), 2012–2026 (2011)
9. T Vaudrey, C Rabe, R Klette, J Milburn, in *2008 23rd International Conference Image and Vision Computing New Zealand, IVCNZ*. Differences between stereo and motion behaviour on synthetic and real-world stereo sequences, (2008), pp. 1–6
10. C Solomon, T Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab* (Wiley-Blackwell, 2010)
11. A Sappa, F Dornaika, D Ponsa, D Gerónimo, A López, An efficient approach to onboard stereo vision system pose estimation. IEEE Trans. Intell. Trans. Syst. **9**(3), 476–490 (2008)
12. G Dubbelman, W van der Mark, J van den Heuvel, F Groen, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*. Obstacle detection during day and night conditions using stereo vision, (2007), pp. 109–116
13. R Labayrade, D Aubert, J Tarel, in *Proceedings of IEEE Intelligent Vehicle Symposium*, vol. 2. Real time obstacle detection on non flat road geometry through 'V-disparity' representation, Versailles, France, 2002), pp. 646–651
14. R Hartley, A Zisserman, *Multiple View, Geometry in Computer Vision* (Cambridge University Press, New York, NY, USA, 2003)
15. Z Zhang, A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
16. Z Zhang, Determining the epipolar geometry and its uncertainty: a review. Int. J. Comput. Vis. **27**(2), 161–195 (1998)

17. T Vaudrey, R Klette, in *Proceedings of the 31st DAGM Symposium on Pattern Recognition*. Residual images remove illumination artifacts! (Springer-Verlag, Berlin, Heidelberg, 2009), pp. 472–481
18. S Birchfield, C Tomasi, Depth discontinuities by pixel-to-pixel stereo. Int. J. Comput. Vis. **35**(3), 269–293 (1999)
19. L Nalpantidis, A Gasteratos, Stereo vision for robotic applications in the presence of non-ideal lighting conditions. Image Vis. Comput. **28**(6), 940–951 (2010)
20. K Zhang, J Lu, G Lafruit, Cross-based local stereo matching using orthogonal integral images. IEEE Trans. Circ. Syst. Video Technol. **19**(7), 1073–1079 (2009)
21. K Yoon, I Kweon, Locally adaptive support-weight approach for visual correspondence search. IEEE Trans. Circ. Syst. Video Technol. **2**, 924–931 (2005)
22. H Hirschmuller, P Innocent, J Garibaldi, Real-time correlation-based stereo vision with reduced border errors. Int. J. Comput. Vis. **47**(1-3), 229–246 (2002)

doi:10.1186/1687-5281-2012-13
Cite this article as: Mroz and Breckon: **An empirical comparison of real-time dense stereo approaches for use in the automotive environment.** *EURASIP Journal on Image and Video Processing* 2012 **2012**:13.

2012-08-16

An empirical comparison of real-time dense stereo approaches for use in the automotive environment

Mroz, Filip

Springer / Hindawi

Mroz F, Breckon T. (2012) An empirical comparison of real-time dense stereo approaches for use in the automotive environment. EURASIP Journal on Image and Video Processing, Volume 2012, Article number 13

https://doi.org/10.1186/1687-5281-2012-13